Energy-efficient Capture of Stochastic Events by Global- and Local-Periodic Network Coverage*

Shibo He
State Key Lab. of
Industrial Control Technology
Zhejiang University
Hangzhou, China
ferrer@zju.edu.cn

Jiming Chen
State Key Lab. of
Industrial Control Technology
Zhejiang University
Hangzhou, China
jmchen@ieee.org

David K. Y. Yau
Dept. of Computer Science
Purdue University
West Lafayette, IN, USA
yau@cs.purdue.edu

Huanyu Shao
Dept. of Computer Science
Purdue University
West Lafayette, IN, USA
shaoh@cs.purdue.edu

Youxian Sun
State Key Lab. of
Industrial Control Technology
Zhejiang University
Hangzhou, China
yxsun@iipc.zju.edu.cn

ABSTRACT

We consider a high density of sensors randomly placed in a geographical area for event monitoring. The monitoring regions of the sensors may have significant overlap, and a subset of the sensors can be turned off to conserve energy, thereby increasing the lifetime of the monitoring network. Prior work in this area does not consider the event dynamics. In this paper, we show that knowledge about the event dynamics can be exploited for significant energy savings, by putting the sensors on a periodic on/off schedule. We discuss energy-aware optimization of the periodic schedule for both cases of a *synchronous* and an *asynchronous* network. Under the periodic scheduling, coordinated sleep by the sensors can be applied orthogonally to minimize the redundancy of coverage and further improve the energy efficiency. We consider four points in the design space: synchronous periodic scheduling with and without coordinated sleep, and asynchronous periodic scheduling with and without coordinated sleep. We show that the asynchronous network exceeds the synchronous network in the coverage intensity, thereby increasing the effectiveness of the event capture, though it may also reduce the opportunities for coordinated sleep. When the sensor density is high, the asynchronous network with coordinated sleep can achieve extremely good event capture performance while being highly energy-efficient.

*Research supported in part by Natural Science Foundation of China (NSFC) under grant no. 60604029, by Joint Funds of NSFC-Guangdong under grant no. U0735003, by 863 High-tech Program no. 2007AA041201, and by 111 Projects under grant no. B07031.

MobiHoc'09, May 18–21, 2009, New Orleans, Louisiana, USA.

Categories and Subject Descriptors

C.2.1 [**Computer-communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-communication Networks**]: Network Protocols

General Terms

Algorithms, Design, Performance

Keywords

Sensor network, network coverage, periodic scheduling, coordinated sleep protocol, event capture, energy efficiency

1. INTRODUCTION

The coverage problem is a fundamental problem in sensor network design. The problem is about the placement and/or scheduling of sensors to maximize the ability to detect or capture interesting events appearing in a deployment area. As technologies advance, practical sensors such as motes and smartdust may have small form factors and low cost. It is feasible to deploy a large number of these sensors for area monitoring. Carefully controlled placements of the sensors may be difficult, due to their large numbers or challenges of the geography. Instead, a loosely controlled method to place the sensors may be used. For example, an airplane is used to drop and scatter sensors to cover a mountainous terrain to detect the presence of animals and identify them. The operation will result in random placements of the sensors. Moreover, unattended operation of the sensor network is often desirable or required, and there is a need to maximize the lifetime of the network before the sensors run out of energy.

To achieve the goal of energy efficiency, there is a need to duty-cycle the sensors to minimize the redundancy of coverage caused by overlap in their sensing regions. In a *coordinated sleep* approach, we determine when a sensor, say S, is made redundant because its sensing region is completely covered by those of its active neighbors. We can then safely turn off S to conserve energy without hurting the performance. Moreover, it is desirable to rotate the active sensors

to achieve energy balance, so that different subsets of the sensors are active at different times. The load balancing prolongs the lifetime of the network before a significant number of the sensors die and cause a severe loss in the coverage.

Prior work in coordinated sleep takes a conservative approach. It tries to ensure that every point of the deployment area is covered all the time by at least one active sensor, provided that there is an available sensor in the random placement. The *role-alternating, coverage-preserving* (RACP) algorithm in [8] is designed to minimize the probability that any given point is not covered by an active sensor, provided that it could be covered. Doing so will also maximize the detection probability of any event and ensure instantaneous detection, if the event is within range of at least one sensor that is alive. In some applications for transient events (e.g., an animal which arrives and then leaves), the goal may be to maximize the detection probability of the events before they disappear, and some delay in the detection is acceptable. Given the relaxed performance objective, leaving an area uncovered some of the time may be acceptable, since an event arriving when there is no active sensor may stay long enough until a sensor becomes active. For simplicity, we can consider periodic scheduling of the individual sensors, in which a sensor is active for only q time every p time ($q \leq p$). It has been shown in [2, 14] that knowledge about the event dynamics—the stochastic processes of the event arrivals/departures—can be used to optimize the periodic schedule of a single sensor for event capture.

Our main contribution in this paper is to expose the interactions between periodic scheduling and coordinated sleep in a dense static sensor network. We will extend the results in [2, 14] to consider energy efficiency and the collective performance of the sensors. The basic observation is that if events may stay for some time, then for a point covered periodically for q time every p time, the fraction of events captured there may be significantly more than q/p. Given the (q, p) schedules of the sensors, the global network itself will achieve the same periodic coverage schedule if the on periods of the sensors are *synchronous* (i.e., they start at the same time). We call this configuration *global-periodic* network coverage. If the on period of each sensor starts at a uniformly random point within p, then the periodic sensor schedules are *asynchronous*. We call this configuration *local-periodic* network coverage. We derive the optimal periodic schedule (q^*, p^*) for both kinds of network. Coordinated sleep by the sensors can then be applied orthogonally to further improve the energy efficiency. We therefore have four design points: (i) synchronous periodic coverage without coordinated sleep (S-nc), (ii) synchronous periodic coverage with coordinated sleep (S-CSP), (iii) asynchronous periodic coverage without coordinated sleep (A-nc), and (iv) asynchronous periodic coverage with coordinated sleep (A-CSP).

Apart from its simplicity, S-nc is mostly interesting as a basis for performance comparison, since its performance is otherwise dominated by that of S-CSP. All the other approaches of S-CSP, A-nc, and A-CSP are of practical interest, and it is instructive to compare their performance with each other and with the RACP protocol in [8]. We have the following findings.

(i) Among the periodic scheduling approaches, S-CSP maximizes the opportunities for coordinated sleep. It can thus achieve the longest network lifetime at the price of some performance loss in event capture. When q is small compared with p, S-CSP is significantly more energy-efficient than RACP, but it is inferior to RACP in terms of capturing events without delay. The performance gap closes significantly, however, when we measure the fraction of events captured with or without delay. S-nc performs better in this measure because it is designed to exploit the possibility of capturing an event before the event leaves.

(ii) A-nc has the least overhead among all the approaches because it requires zero sensor coordination. It has the same network lifetime as S-nc (hence shorter lifetime than S-CSP) but it has better event-capture performance than the synchronous approaches, by spreading out the redundant coverage for a higher global coverage intensity. Under a wide range of q/p, its performance in terms of event capture (with or without delay) is extremely competitive with that of RACP and it has a longer network lifetime than RACP. When the sensor density is high, it is extremely competitive with RACP in terms of *instantaneous* event capture also, while remaining more energy efficient than RACP. Hence, periodically resting the sensors to conserve energy does *not* necessarily add noticeable delay in the event capture.

(iii) A-CSP achieves the same event-capture performance (instantaneous or delayed) as A-nc (hence the same competitive performance with RACP), but it can further extend the network lifetime beyond A-nc. When the sensors' periodic schedules are asynchronous, however, the chance for coordinated sleep decreases. A higher sensor density will then help A-CSP realize its potential for energy savings, provided *also* that q is large enough relative to the energy costs of turning off/on the sensor. Our performance results show that *if the sensor density is high and both instantaneous and delayed capture performance are important, A-CSP represents the overall best method.*

2. RELATED WORK

Sensor coverage problems have generally considered two deployment scenarios. In a sparse network, the sensor density is low and there is little redundancy of coverage. The problem is to optimize the placement of the sensors for maximum coverage or k-coverage under different performance objectives. In a dense network, the coverage regions of the sensors have significant overlap. The problem is to duty-cycle the sensors to achieve area coverage or area k-coverage while minimizing the use of redundant resources. Dynamically load balancing the sensors by their residual energies is relevant for maximizing the network lifetime.

In this paper, we are concerned with the coordinated deployment of a high density of randomly placed sensors in a geographical area. The sensors implement a coordinated sleep algorithm to eliminate the redundancy of coverage while balancing their energy use. The coordinated sleep approach has been studied in prior work [8, 10, 11, 13]. A centralized off-line algorithm has been proposed by Slijepcevic and Potkonjak [10]. Their approach determines a number of k-covers of sensors at the beginning, so that the sensors in any one cover can be active at a time to provide area coverage. The offline approach cannot react to the loss of sensors due to depleted energies or other contingencies. In a decentralized on-line approach, Hsin and Liu [8] have compared the performance between random sleep and

coordinated sleep, and shown that coordinated sleep can result in significantly more energy savings. Our work uses a similar decentralized on-line algorithm. However, the prior approaches [8, 11, 13] do not consider the event dynamics in the network design. We recognize that the event dynamics can be exploited for significant energy savings, by putting the sensors on a periodic on/off schedule, after optimizing the schedule for maximum event capture per unit of energy use. We show that interesting performance tradeoffs arise in the interactions between periodic scheduling and coordinated sleep.

Optimizing the network design using knowledge about the dynamics of stochastic events has been studied in [2]. They analyze the best movement strategy of a sensor circulating between a number of points of interest (PoIs) to maximize a *quality of capture* (QoC) metric. The QoC is defined as the expected capture rate of events, with or without the constraint that a minimum rate of capture must be satisfied at *every* PoI. If the maximum speed of a single sensor does not support the required QoC, they also analyze the minimum number of sensors necessary. Their problem is concerned with one or a few mobile sensors which do not coordinate with each other during deployment. We study a high density of static sensors which may coordinate their work schedules. The QoC analysis in [2] is generalized in [14], where the authors consider a *quality of monitoring* (QoM) metric defined in terms of captured *information*. An event utility function specifies the rate of information gained as a function of the sensing time, and they consider general forms of this function. Our performance metric of the number of events captured can be extended readily to consider their information metric. Their work considers the periodic scheduling of individual sensors but, similar to [2], they do not consider energy efficiency or the coordination issues between sensors, in particular, how the individual sensor schedules may combine to give different schedules of the global network.

Connectivity is a complementary problem to coverage. We do not address connectivity in this paper except to note that duty cycling sensors has the benefit of reducing contention in the data reporting. There is a lot of existing work on the relationship between coverage and connectivity [1, 9, 12, 15].

3. PROBLEM SETUP AND PERFORMANCE METRICS

We assume that sensors are used to cover an area for capturing stochastic events. We use the Boolean model of sensing, meaning that an event is *captured* if its distance from an active sensor is less than distance r, where r is the sensing range. The Boolean model is widely used in the literature due to its simplicity. While more complex anisotropic models [7] could be used for increased accuracy, doing so will not change our conclusions qualitatively. We assume that sensors can be turned off (made inactive) independently to conserve energy. Strictly speaking, each sensor has three main functional modules—sensing, communication, and computation—which can be turned off independently. For simplicity we will assume that when the sensor becomes inactive, all three functional modules are inactive. According to specifications of real-world sensors such as Crossbow motes [3], the sensor in the active and inactive states consumes energy at rates of k_1 and k_2, respectively, where k_1 and k_2 are constants. Moreover, a constant amount of energy, given by c, is needed for the sensor to change between the active/inactive states.

We assume that transient random events appear at given *points of interest* (PoIs). The average rate of event arrivals at a PoI is given by λ. After an event arrives at PoI i, it stays for a random *event staying time* drawn from a distribution X. After that, the event disappears and after another random *event absence time* drawn from a distribution Y, the next event arrives at i. We assume that the staying time and the subsequent event absence time of different events are i.i.d., even though the two quantities for the same event may be dependent. We do not exploit the spatial correlation of PoIs in the network design. Hence, we assume that the event dynamics at different PoIs are independent.

We consider random placement of the sensors according to a Poisson point process of intensity γ. The Poisson point process is widely used in the analysis of random node placements in networks [5]. When γ is large, there is significant redundancy between the sensing regions of sensors. Strictly speaking, a sensor S is made redundant by its active neighbors if there is no PoI in the sensing region of S that is not covered by those of the neighbors. We assume, however, that the PoIs are spatially located at a high and uniform density (e.g., there are many evenly placed PoIs within the sensing region). Hence, we will adopt a more simple and conservative redundancy test: A sensor S is made redundant by its active neighbors if its sensing region is completely covered by those of the neighbors.

In quantifying the performance of the sensor network, we restrict our attention *only* to PoIs that are within distance r of at least one sensor. It is because if events happen at PoIs not within range of any sensor, the events cannot be captured irrespective of the network design,[1] and we consider these events out of scope. We use two principal performance metrics: (i) the probability of instantaneous capture, P_{in}, which is the probability that an event arriving at a PoI will be captured immediately upon arrival (without delay); and (ii) the probability of capture, P_c, which is the probability that an event appearing in a PoI will be captured before it leaves the PoI (though the event does not necessarily leave the network). It is clear that $P_c \geq P_{in}$. In many applications, some delay in the detection is acceptable. In a wildlife monitoring network, for example, researchers may be interested in identifying the animals that pass by a given point, but they do not need immediate report of the information. Also, when events are detected with delays, the average delay can be further quantified for performance evaluation.

4. EVENT CAPTURE BY PERIODIC SENSOR

We now analyze the per-PoI event capture performance of a sensor on a periodic schedule (q, p). The periodic schedule means that before energy runs out, the sensor is alternately active and inactive for q time and $p - q$ time, respectively. The following theorem concerns the probability that an event is captured by the sensor and is a paraphrase of Theorem 2 in [14].

THEOREM 1. *For a (q, p) periodic sensor whose sensing region covers PoI i, the sensor captures an event at i (before*

[1] Since we do not control the sensor placement.

the event disappears) with probability

$$P_c = \frac{q}{p} + \frac{1}{p}\int_0^{p-q} \Pr(X \ge t)\,dt$$

before it runs out of energy.

Proof. Refer to [14], Theorem 2. □

Note that the expression for P_c is a sum of two terms. The first term gives the fraction of events captured during the sensor presence period $[0, q]$. It is equivalent to the performance measure P_{in} since these events are captured instantaneously. The second term gives the fraction of captured events that arrive during the sensor absence period $[q, p]$, but stay long enough to be captured during the *next* sensor presence period $[p, p+q]$. These events are captured with a delay, and therefore contribute to the performance measure P_c. The main observation is that due to the contribution of the second term, the sensor working for q time every p time may capture a fraction of events that is much higher than q/p. We intend to exploit this property of the periodic sensor to achieve high energy savings with relatively minor loss in the capture performance.

For those events captured with a delay, we can analyze the average delay given by the following theorem.

Theorem 2. *For the events captured at PoI i that arrive during a sensor absence period, the expected delay until their capture at i is given by*

$$D = \frac{\int_0^{p-q} \Pr(X \ge t) \times t\,dt}{\int_0^{p-q} \Pr(X \ge t)\,dt}.$$

Proof. An event arriving t time before the next sensor presence period will be captured with delay t provided that it stays long enough. Since the event must arrive during the current sensor absence period, the value of t ranges from 0 to $p - q$. The numerator times λ is the total delay of all the captured events, where λ is the average rate of event arrivals. The denominator times λ gives the number of captured events. The result follows as the average delay per captured event. □

Theorem 1 applies to general distributions of the event staying times. We can illustrate the result using the Exponential distribution with rate parameter α, i.e., the pdf of X, denoted by $f(x)$, is given by:

$$f(x) = \alpha e^{-\alpha x}, \quad x > 0, \quad \text{mean} = \frac{1}{\alpha}.$$

Then we have

$$P_c[X \in \text{Exponential}(\alpha)] = \frac{q}{p} + \frac{1 - e^{-\alpha(p-q)}}{p\alpha}. \tag{1}$$

Similarly, the D value in Thoerem 2 specializes to

$$D[X \in \text{Exponential}(\alpha)] = \frac{1 - e^{-\alpha s}(\alpha s + 1)}{\alpha(1 - e^{-\alpha s})}, \tag{2}$$

where $s = p - q$.

5. ENERGY-AWARE OPTIMIZATION OF SYNCHRONOUS PERIODIC NETWORK

We discuss optimization of the periodic schedule (q, p) for the *synchronous* network. In such a network, all the sensors employ the same (q, p) schedule, and they start their on periods at the same time so that the on/off periods are synchronized. Thus, the global network as a whole behaves like one big sensor on the (q, p)-periodic schedule. In practice, lightweight and accurate time synchronization protocols are available [6] to support the implementation of the synchronous network.

It can be shown that, for general distributions of X, the P_c value in Theorem 1 has two properties: (i) for the same p, P_c is monotonically increasing in q/p; (ii) for the same q/p, P_c is monotonically decreasing in p. Hence, to optimize the event capture, we can either make $\frac{q}{p} = 1$, or if we decide to use a smaller q/p, we can make p as small as possible. The first option will not help us save energy, while the second option is limited physically by the delay and energy expense in switching the sensor between the on/off states frequently.

To optimize the periodic schedule, we need to explicitly account for energy use by the energy model given in Section 3.[2] Specifically, we know that for a (q, p)-periodic sensor, it is able to capture a fraction P_c of events according to Theorem 1. Hence, its per-PoI rate of capturing events is given by $Q' = \lambda \times P_c$. On the other hand, the rate of energy use of such a sensor is given by

$$E' = \frac{k_1 \cdot q + k_2(p - q) + 2c}{p}. \tag{3}$$

We allow the user to specify the minimum P_{in}, denoted by $P_{in}^{\min}$, that events are detected without delay, where $P_{in}^{\min} > 0$. By Theorem 1, P_{in} is equal to q/p. Hence, we set $\frac{q^*}{p^*} = P_{in}^{\min}$ and optimize P_c by solving the following optimization problem for the *per-PoI number of events captured per unit of energy*, denoted by Q_E:

$$\text{Find } p^* = \arg\max_p \; Q_E = \arg\max_p \; Q'/E'.$$

(Note that the above formulation is equivalent to optimizing the expected number of events captured before an energy budget given by B is depleted.) For example, if $X \in$ Exponential(α) and $Y \in$ Exponential(β), then $\lambda = \alpha\beta/(\alpha + \beta)$. By (1), the rate of capturing events at a PoI is

$$Q' = \frac{\beta}{p(\alpha + \beta)}(1 + q\alpha - e^{-\alpha(p-q)}). \tag{4}$$

The optimization problem is then to find p^* that maximizes

$$Q_E = \frac{\beta}{\alpha + \beta} \frac{1 + q\alpha - e^{-\alpha(p-q)}}{k_1 \cdot q + k_2(p - q) + 2c}, \tag{5}$$

where $q = P_{in}^{\min} \times p$. Note that we have a one-dimensional optimization problem, and the solution can be computed numerically by comparing the points of p where

$$\frac{dQ_E}{dp} = 0 \text{ and } \frac{d^2 Q_E}{dp^2} < 0.$$

Remark. Note that in the network, more than one PoI may be within range of a sensor, and the same PoI may be within range of more than one sensor. Hence, the sensors may not capture events that are distinct. Assume that there are m distinct PoIs within range of n sensors in the network. The number of distinct events captured per unit energy for the whole network is $\frac{m \times Q'}{n \times E'}$. This value is the Q_E value derived

[2]For simplicity, we do not consider the latency constraints of turning on/off the sensor, but note that such constraints can be easily incorporated.

above scaled by a constant factor of $\frac{m}{n}$. The scaling by a constant factor will *not* affect the solutions to the optimization problem.

6. OPTIMIZATION OF ASYNCHRONOUS PERIODIC NETWORK

We now analyze the *asynchronous* network. In this kind of network, each sensor employs the same (q, p)-periodic schedule, but they start their on periods independently at a uniformly random point in time within the period p. Because the on periods of the sensors are spread out in the asynchronous network, the event capture performance must consider the joint operation of the sensors. We have the following main results (Theorems 3 and 4).

THEOREM 3. *For a random placement of sensors by the Poisson point process of intensity γ, the probability that an event appearing at a PoI is captured instantaneously is given by $P_{in} = \frac{1-e^{-\gamma\pi r^2 \frac{q}{p}}}{1-e^{-\gamma\pi r^2}}$, where r is the sensing range.*

PROOF. A sensor can potentially detect an event if the event happens within distance r of the sensor. Hence, an event may be potentially detected by any sensor within the circular region centered at the event and of radius r. Note that at least one such sensor exists, by the definitions of our performance metrics P_{in} and P_c. By the property of the Poisson point process, the probability p_k that there are k sensors in the circular region is given by

$$p_k = \frac{(\gamma\pi r^2)^k e^{-\gamma\pi r^2}}{k!}. \tag{6}$$

The probability that any sensor is inactive when the event happens is $1 - \frac{q}{p}$. The event is undetected on arrival only if all the k sensors are inactive, which happens with probability $(1 - \frac{q}{p})^k$. Hence, summing over the range of k, we can compute the probability that the event is undetected on arrival as

$$\frac{1}{1-p_0}\sum_{k=1}^{\infty}(1-\frac{q}{p})^k\, p_k = \frac{e^{-\gamma\pi r^2 \frac{q}{p}} - p_0}{1-p_0}.$$

The result follows as the complement of the above probability. ☐

Remark. Whereas $1 - P_{in}$ decreases linearly with $\frac{q}{p}$ in the synchronous network, the decrease is exponential in the asynchronous case. This implies that for the asynchronous network, a small increase in $\frac{q}{p}$ may result in a large increase in P_{in}. We will show in Section 8.2 that the asynchronous network can indeed achieve high energy efficiency with an extremely small loss in event capture performance.

THEOREM 4. $[X \in \text{Exponential}(\alpha)]$ *For a random placement of sensors by the Poisson point process of intensity γ, the probability that an event appearing at a PoI is captured before it leaves the PoI is given by $P_c = \frac{1-e^{\gamma\pi r^2(\rho-1)}}{1-e^{-\gamma\pi r^2}}$, where r is the sensing range and $\rho = \frac{\alpha(p-q)-1+e^{-\alpha(p-q)}}{p\times\alpha}$.*

PROOF. Let $s = p - q$. Consider a sensor in the circular region of radius r centered at the event. The sensor starts its most recent off period at time 0. Consider an event arriving at time t and staying for time x, where $x \sim X \in \text{Exponential}(\alpha)$. The sensor does not detect the event if it is inactive in $[t, t + x]$, which occurs with probability $\Pr(x < s - t)$ provided that $t < s$. Hence, summing over the range of t and denoting by ρ the probability that the sensor does not detect the event before the event leaves, we have

$$\rho = \frac{1}{p}\int_0^s 1 - e^{-\alpha(s-t)}\, dt = \frac{\alpha s - 1 + e^{-\alpha s}}{p\times\alpha}.$$

For the Poisson point process, (6) gives the probability that there are p_k sensors within range of the event. The probability that the event is undetected by any in-range sensor before it leaves is therefore

$$\frac{1}{1-p_0}\sum_{k=1}^{\infty} p_k\, \rho^k = \frac{e^{\gamma\pi r^2(\rho-1)} - p_0}{1-p_0}.$$

The result follows as the complement of the above probability. ☐

As with the synchronous network, we allow the user to specify the minimum probability $P_{in}^{\min}$ of instantaneous event detection. Notice that P_{in} is montonically increasing in both the sensor density γ and the ratio q/p. When $q = p$, $P_{in} = 1$. Hence, any P_{in} is satisfiable. We can compute the smallest q/p needed to satisfy $P_{in}^{\min}$ and denote this value by z. Once z is determined, we can find p^* (hence $q^* = z\, p^*$) as the optimal p that maximizes $Q_E = \frac{P_c\times\lambda}{E'}$. As in the synchronous network case, we have a one-dimensional optimization problem, although the expression for P_c is more complex.

Note also that if we considered the capture probabilities of *all* events (i.e., whether they fall within range of at least one sensor or not), the above Q_E measure would be scaled by $1 - p_0$, which is independent of p and q and hence will not affect the optimization problem.

7. COORDINATED SLEEP UNDER PERIODIC SCHEDULING

In a dense network, there is significant spatial overlap in the sensing regions of sensors. Temporally, the on periods of the periodic sensors may also overlap. The latter problem is the most severe in the case of the synchronous network, but it is also a concern in the asynchronous network particularly when q/p is large. Such overlap leads to coverage redundancy and waste of energy. When a significant fraction of the sensors die, the coverage of the network degrades severely.

We propose a solution in which sensors exchange information about their locations, residual energies, etc, so that a sensor, say i, whose sensing region is completely covered by those of its active neighbors can go to sleep to conserve energy. Before i can go to sleep, it needs permission from its neighbors because they have to agree to remain active and cover i's responsibilities. Sensors renegotiate the permission to sleep from time to time. The decision to grant permission is based on the residual energies of the competing nodes, which allows sensors to rotate their roles and achieve an energy balance for maximum network lifetime.

We now present a *coordinated sleep protocol* (CSP) for sensors to achieve the goals above. The protocol can be applied to both synchronous and asynchronous networks. CSP is similar to the RACP protocol in [8], but our main purpose is to study its use in a periodic scheduling context. In Section 8, we will evaluate how the two techniques interact in

four design points. We will also present performance comparisons with RACP where periodic scheduling can achieve significant energy savings with competitive performance.

In CSP, an active sensor can assume one of three roles: *regular*, *supporting*, and *redundant*. Sensors periodically exchange hello messages indicating their ids, locations, period start times, roles, and residual energies.[3] The locations of the sensors can be obtained by GPS or a localization protocol [4]. A supporting sensor has agreed to be active for a stated amount of time to support the sleep of a neighbor. While active in supporting role, the sensor maintains a timer that signals the end of this role. When the timer expires, the sensor changes role to regular.

For a regular sensor, say i, each subset of its active neighbors whose sensing regions completely cover that of i is called a *support* of i. Sensor i periodically checks if it has a non-empty support set. If so, i rates a support by (i) the minimum residual energy of its members, and (ii) the overlap, denoted by L, between the intersection of the members' on periods and i's own on period. Sensor i prefers supports with high minimum residual energy but in case two supports are close by that metric, it may select the one with a longer L. It then broadcasts a request-to-sleep (RTS) packet to its neighbors in the chosen support. Each neighbor who does not desire to go to sleep immediately replies to i with a clear-to-sleep (CTS) packet. Once i receives a CTS from all the support's members, it broadcasts a confirm (CNF) packet to the support, changes to redundant role, and sleeps for L time, after which it changes back to regular. While still waiting for at least one more CTS, i may receive an RTS from a neighbor, say j. In this case, i sets a (per-neighbor) random delay timer T_j for j. If, by the time T_j expires, i is still waiting for a CTS, i sends j a CTS. Once i receives a CNF from j, it assumes supporting role for j for the specified time.

Note that if L is less than $\frac{2c}{k_1 - k_2}$, the sleep is too short to be productive for energy savings. In this case, i will simply decide not to send an RTS. This situation can be common if q is small (particularly for asynchronous networks), so that the temporal redundancy is not clear enough for coordinated sleep to help even if the sensor density is high. Another issue is i's setting of the random delay T_j. The design principle is that if i has a larger residual energy (e_i) than that of j (e_j), then i should be more likely to send j the CTS before j sends i the CTS. Hence, it should be likely that i's delay timer for j is smaller than j's timer for i. We choose to pick T_j uniformly at random from $[0, 2^{\frac{e_j}{e_i}}]$, but alternative methods are possible.

Apart from support for periodic scheduling, CSP differs from RACP in its use of the delay timer. In RACP, a countdown delay is mandatory for every RTS. In CSP, i sends an RTS without delay, but a neighbor j who receives the RTS performs the random countdown if (and only if) j also desires to go to sleep. Hence, a countdown delay is avoided in CSP if there is no competition between neighbors in their sleep requests. Also, RACP uses a random sleep duration after a sensor has obtained permission to sleep, whereas CSP uses L as determined by the overlapping on periods of the sensors. Our experiments show that RACP's performance is more random in terms of variable network lifetimes achieved, while the lifetime of CSP is much more predictable.

[3]If the on periods of two sensors do not overlap, they do not hear each other's hello messages. This does not affect performance since there is no temporal overlap of coverage between the two sensors.

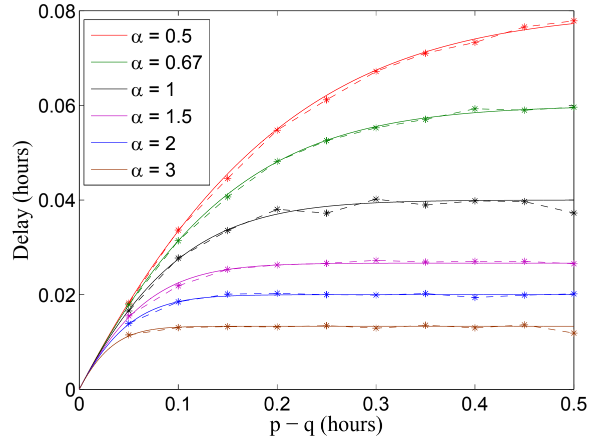

Figure 1: For events captured with positive delay, the average delay against $p-q$ for varied α.

8. NUMERICAL RESULTS

We present Matlab results to illustrate and verify the analysis. In addition, we present Matlab simulations to evaluate the performance of the synchronous and asynchronous networks with and without coordinated sleep, including performance comparisons with RACP.

Unless otherwise stated, we use the following: (i) $X \in$ Exponential(α), $Y \in$ Exponential(β), $\alpha = 1$, and $\beta = 2$, where the distributions are numbers in time units of 0.1 h; (ii) for the energy model, $k_1 = 2.369$ J/h, $k_2 = 0.17$ J/h, and $c = 0.05$ J; (iii) each sensor of sensing range $r = 1$ m has an energy capacity of 9.26 mAh (equivalent to 100 J assuming a voltage of 3 V), and the sensors are deployed in a 20 m $\times$ 20 m region according to a Poisson point process of intensity $\gamma = 4$, where γ is the average number of sensors per m^2; and (iv) for the distribution of PoIs, the deployment region is discretized into cells of dimensions 0.2 m $\times$ 0.2 m and the center of each cell is a PoI. Each simulation in Section 8.1 is repeated until the standard deviation of the measurements is negligible compared with the average, and we report the average in the presentation. Representative traces are reported in Section 8.2.

8.1 Illustration of analytical results

A) Average delay of capture. Our approach allows a $P_c - P_{in}$ fraction of events to be captured with positive delay. For these events, Theorem 2 and Eq. (2) quantify the average delay. Fig. 1 plots the analytical results against $p-q$ for different α. The measured averages in simulations are also shown as the indicated data points. Notice that the delay increases as the mean event staying time $\frac{1}{\alpha}$ increases. Also, when $p-q$ is large enough, most events that arrive early in an absence period do not stay long enough to be captured. Hence, the delay does not further increase with $p-q$ after some point. The quantified average delay can be used as an additional constraint in our network optimization, although we do not consider it in this paper.

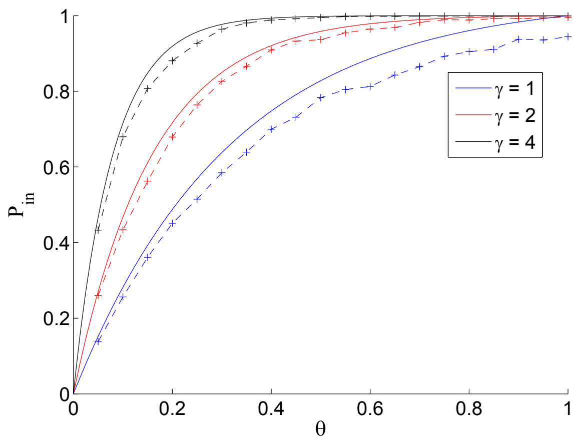

Figure 2: Plot of P_{in} of asynchronous network against $\theta \triangleq \frac{q}{p}$ for different γ.

B) P_{in} of asynchronous network. Fig. 2 verifies the correctness of Theorem 3 by plotting P_{in} against q/p for different γ. Notice the close agreement between the analytical and simulation results. More importantly, notice that at high sensor density (e.g., $\gamma = 4$), the achieved P_{in} is extremely close to 1 even if there is a significant fraction of off time in the periodic schedule (e.g., $\frac{q}{p} = 0.4$). *This shows that periodically resting the sensors does* not *necessarily cause noticeable delay in the event capture.*

C) Q_E of synchronous network. Fig. 3 plots P_c (Theorem 1) against p for different values of q/p. Data points from simulations are also shown, which are in strong agreement with the analysis. For a fixed q/p, the function is monotonically decreasing in p, showing that we should turn on the sensor as briefly as possible at the start of each period. This is because once the sensor detects an event, the event is considered captured and the sensor does not gain by observing the event any longer. When energy is also considered, however, the Q_E plots given by (5) (Fig. 4) initially increase with p and then decrease after reaching a global maximum. (The validation of Q_E follows directly from the validation of P_c.) Hence, the method outlined in Section 5 can be used to find the optimal p that maximizes Q_E.

D) Q_E of asynchronous network. We now illustrate Q_E, computed via the expression for P_c in Theorem 4, for the asynchronous network. We plot Q_E against p in Fig. 5 for different q/p. Notice that the plots are similar in trend to the Q_E plots of the synchronous network (Fig. 4), although the peak performance is much less pronounced in this case. In particular, when $\frac{q}{p}$ is large (e.g., $\frac{q}{p} \geq \frac{1}{3}$), the Q_E value remains close to the optimal as p increases further.

8.2 Network simulations

A) Asynchronous network. We evaluate the asynchronous network with and without coordinated sleep by CSP. We assume that the user desires a stringent $P_{in}^{\min}$ close to 1, so that the required $\frac{q}{p} = 0.4$ by Theorem 3. Given $\frac{q}{p} = 0.4$, $p^* = 0.4$ h by optimizing Q_E numerically according to (5). Hence, $q^* = 0.16$ h.

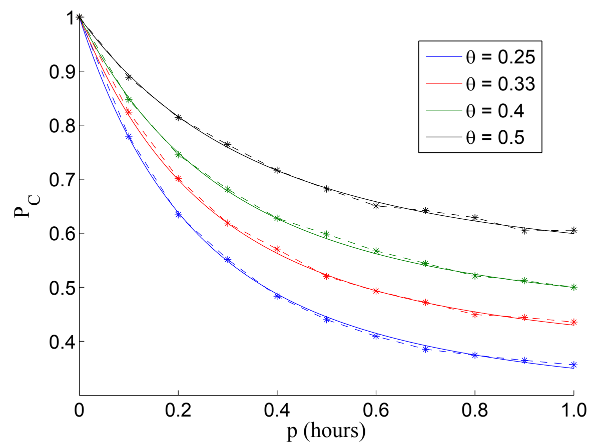

Figure 3: Monotonically decreasing P_c of synchronous network against p for different $\theta \triangleq \frac{q}{p}$.

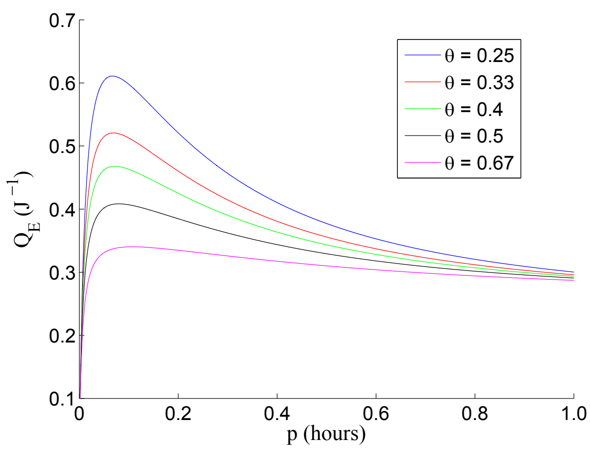

Figure 4: Plot of Q_E of synchronous network against p for different $\theta \triangleq \frac{q}{p}$. Optimal Q_E is achieved at an intermediate p.

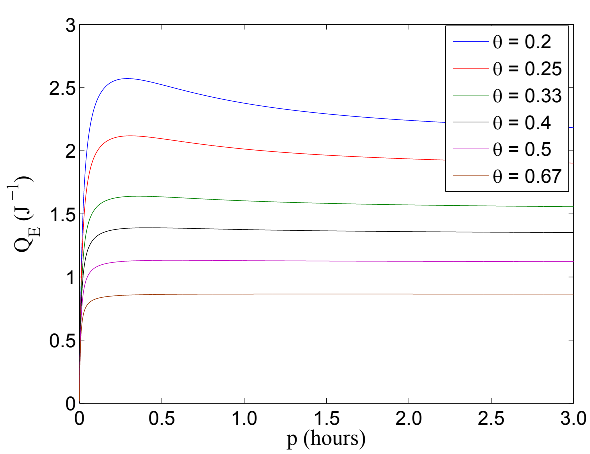

Figure 5: Q_E of asynchronous network against p for different $\theta \triangleq \frac{q}{p}$. Peak of Q_E is less pronounced than synchronous network.

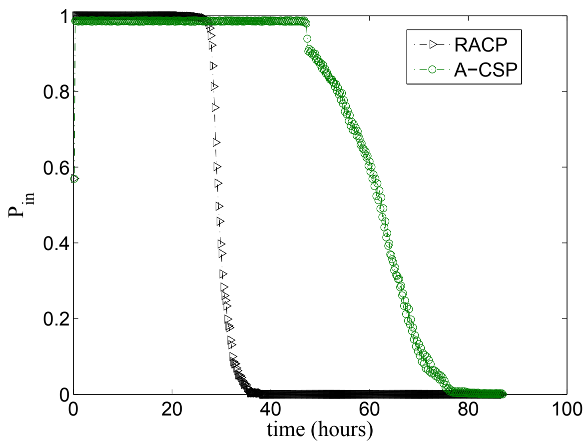

Figure 6: Achieved P_{in} against deployment time for RACP and A-CSP. A-CSP has longer lifetime and more gradual death than RACP.

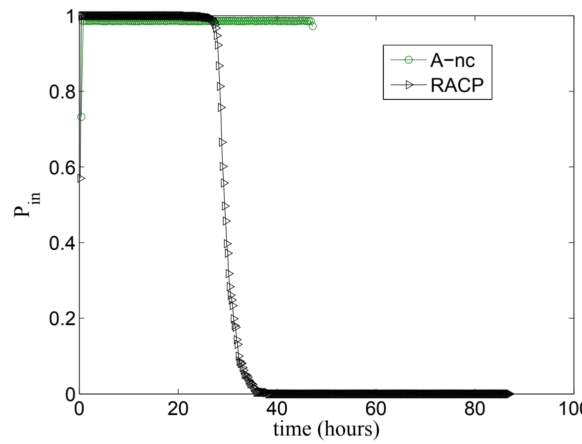

Figure 7: P_{in} *vs.* deployment time for RACP and A-nc. A-nc runs longer than RACP. Its death is instantaneous since all the sensors work equally hard and run out of energy simultaneously.

As a representative trace, Fig. 6 shows the achieved P_{in} as a function of the deployment time for RACP and A-CSP. Notice that A-CSP has quite similar performance as RACP before either network dies, but A-CSP has about 74% longer network lifetime. When the RACP network dies, the coverage drops quickly to zero. Death of the A-CSP network is much more gradual. After A-CSP starts dying, it takes about 30 h more before the network dies completely. Plots of P_c for the two networks are very similar to the corresponding P_{in} plots.

Fig. 7 shows the achieved P_{in} for RACP and the asynchronous network *without* coordinated sleep (i.e., A-nc case). In this case, the asynchronous network has quite similar performance as RACP before either network dies, but it lasts about 74% longer than RACP. Hence, A-nc lasts about as long as A-CSP before A-CSP starts to die. When A-nc dies, however, the coverage immediately drops to zero. This is because all the sensors work equally hard and run out of energy at the same time. Therefore, the usefulness of CSP in the asynchronous network lies mainly in its graceful degradation, where partial (but decreasing) coverage remains available over a significantly longer time duration. Because P_{in} is close to 1 for both networks, performance in terms of P_c is practically the same as P_{in}.

B) Synchronous network. We now evaluate the synchronous network. We assume that the user is willing to relax the requirement for instantaneous event capture, and set $P_{in}^{\min} = 0.4$. Hence, $\frac{q}{p} = 0.4$. Optimizing p for Q_E in (5) yields $P^* = 0.075$ h. Hence, $q^* = 0.03$ h. Fig. 8 compares RACP with S-CSP in terms of P_{in}. Notice that the network lifetime of S-CSP is about 2.1 times that of RACP. However, RACP has maximum instantaneous event-capture performance before it dies, whereas S-CSP achieves the relaxed P_{in} specified by the user (0.4). The performance comparison between RACP and S-CSP in terms of P_c is shown in Fig. 9. Note that S-CSP has performance fluctuating between 0.81 and 0.85, which reflects the periodic on/off schedule of the network. Hence, the performance gap with RACP closes significantly in terms of P_c. This is because the synchronous network is designed to exploit the possibility of capturing

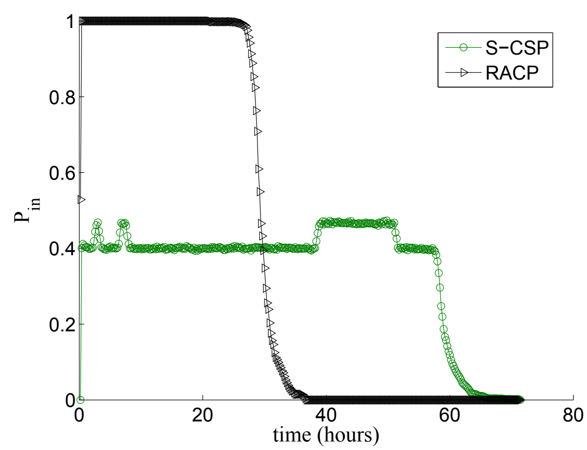

Figure 8: P_{in} against deployment time for S-CSP and RACP. S-CSP achieves $P_{in}^{\min} = 0.4$ as specified by the user, and runs longer than RACP.

an event before the event leaves a PoI. Hence, S-CSP allows the tradeoff of performance (17% lower for P_c) for longer network lifetime (more than 2 times as long).

To evaluate the impact of coordinated sleep on the synchronous network, we compare the performance of S-CSP and S-nc in terms of P_c. The results are shown in Fig. 10. Notice that (i) S-CSP lasts about 34% longer than S-nc before S-CSP starts to die, and (ii) the death of S-nc is abrupt since the sensors work equally hard and they run out of energy at the same time, whereas the death of S-CSP is relatively more gradual.

C) Synchronous/asynchronous network comparison. We now compare the synchronous and asynchronous networks under coordinated sleep. For A-CSP, we use the same evaluation case as in Section 8.2.A, i.e., $P_{in}^{\min}$ close to 1 and $\frac{q}{p} = 0.4$. For S-CSP, we either (i) keep $P_{in}^{\min}$ close to 1, in which case $\frac{q}{p}$ is also close to 1 (since $P_{in} = \frac{q}{p}$ in the

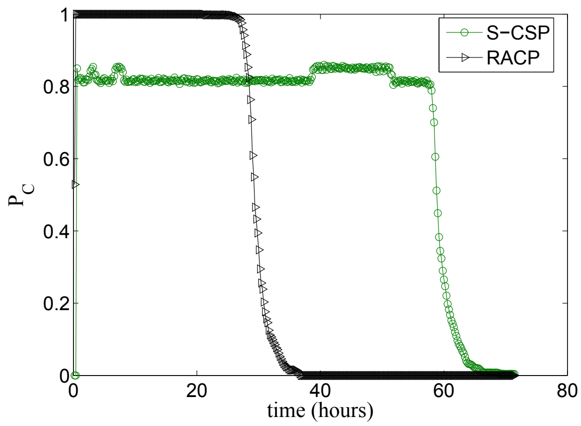

Figure 9: P_c *vs.* deployment time for S-CSP and RACP. S-CSP achieves $P_c = 0.8$ (cf. $P_{in} = 0.4$) and performs closer to RACP in terms of P_c.

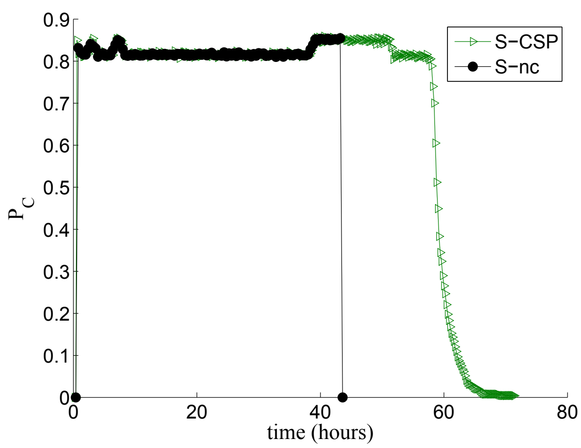

Figure 10: P_c against deployment time for S-CSP and S-nc. CSP prolongs the network lifetime and achieves good energy balance between the sensors (i.e., they die about the same time) in the synchronous network.

synchronous network); or (ii) we keep $\frac{q}{p} = 0.4$, in which case we can only satisfy $P_{in}^{\min} = 0.4$. In case (i), the performance of S-CSP is similar to RACP's (although as Section 7 observes, the network lifetimes of RACP are more variable than S-CSP over different runs) since the sensors are active almost all the time. Hence, A-CSP performs better than S-CSP, as it performs better than RACP. In case (ii), it is clear that A-CSP performs better than S-CSP in terms of P_{in}. To see how A-CSP and S-CSP compare in terms of P_c and the network lifetime, refer to Fig. 11. Note that S-CSP keeps the network running longer before it starts to die and the death is more abrupt, showing that S-CSP can achieve a better energy balance between the sensors. A-CSP starts dying sooner but takes a longer time to reach complete death. Considering the event capture performance throughout the deployment, A-CSP performs better than S-CSP overall.

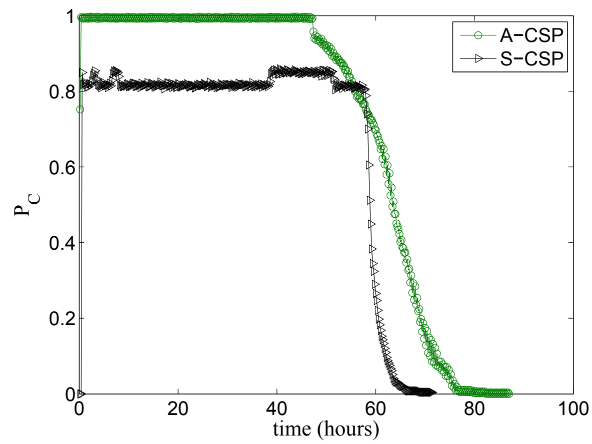

Figure 11: P_c against deployment time for A-CSP and S-CSP. S-CSP operates longer before it starts to die and its death is less gradual than A-CSP. However, A-CSP performs better in terms of capture performance.

8.3 Summary of experiments

The experiments in Section 8.1 verify the analysis in Sections 4–6 and illustrate the optimization of Q_E. The experiments in Section 8.2 compare S-nc, S-CSP, A-nc, A-CSP, and RACP in a dense sensor network. We show that coordinated sleep in S-CSP can prolong the network lifetime compared with S-nc. S-CSP provides a performance/energy tradeoff versus RACP. It achieves the P_{in} specified by the user, and its achieved P_c can be significantly higher than P_{in} for events that stay. For events captured with positive delay, the average delay can be quantified and could be used in the network optimization. We show that asynchronous periodic scheduling can achieve a higher global coverage intensity than S-CSP by spreading out the sensors' on periods. Importantly, the global operation can allow it to achieve maximum performance even in terms of *instantaneous* event capture (i.e., P_{in} close to 1) while being much more energy-efficient than RACP. A-CSP can further prolong the network lifetime over A-nc. It can achieve significantly longer network lifetimes than RACP with negligible loss of performance. A-CSP appears to be the best overall method in terms of maximum network lifetime and extremely good event capture performance.

9. CONCLUSIONS

We have investigated the use of periodic sensor scheduling for capturing stochastic events. We started off with the observation that when events can stay for some time, a (q, p)-periodic sensor can capture a fraction of events much higher than q/p. This is possible because events that arrive when a sensor is inactive may still be captured with a delay. We thus expect a tradeoff between energy efficiency (e.g., smaller q/p) and event capture (e.g., capture delay). We verify that such a meaningful tradeoff exists in the synchronous network relative to RACP. A similar tradeoff exists in the asynchronous network when the sensor density is moderately high, but the tradeoff becomes more attractive because the probability of non-instantaneous event capture

decreases exponentially with $\frac{q}{p}$ in the asynchronous case (cf. linear decrease in the synchronous case). When the sensor density is high, the tradeoff becomes unnecessary in that the asynchronous network can achieve P_{in} close to 1 at high energy efficiency. We have also evaluated the effectiveness of CSP for further energy savings in the synchronous and asynchronous networks. Our results verify that when CSP is applied to a synchronous network, it can significantly increase the network lifetime. CSP is also useful for the asynchronous network. But it can become relatively less effective because the spread-out on periods reduce the chance for coordinated sleep.

Our system model is such that an event is considered fully captured whenever it falls within range of an active sensor. Such a model is widely used in the literature [2, 8], and our results give understanding of some basic properties of periodic scheduling and their interactions with coordinated sleep. By considering different explicit forms of the *event utility function* [14], which quantifies how a non-negligible sensing time is needed to achieve various quality levels of the sensing results, we can further model the *temporal dimension* of a wide range of real-world monitoring tasks.

10. REFERENCES

[1] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *ACM MobiHoc*, 2006.

[2] N. Bisnik, A. Abouzeid, and V. Isler. Stochastic event capture using mobile sensors subject to a quality metric. In *ACM MobiCom*, 2006.

[3] Crossbow, http://www.xbow.com. *Crossbow MPR/MIB Users' Manual*.

[4] D. Evans and L. Hu. Localization for mobile sensor networks. In *ACM MobiCom*, 2004.

[5] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran. Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Trans. Information Theory*, 2007.

[6] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *ACM SenSys*, 2003.

[7] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *ACM Trans. Embedded Comput. Syst.*, 2005.

[8] C. F. Hsin and M. Liu. Network coverage using low duty-cycled sensors: random and coordinated sleep algorithms. In *ACM/IEEE IPSN*, 2004.

[9] R. Iyengar, K. Kar, and S. Banerjee. Low-coordination topologies for redundancy in sensor networks. In *ACM MobiHoc*, 2005.

[10] S. Slijepcevic and M. Potkonjak. Power efficient organization of wirless sensor networks. In *IEEE ICC*, 2001.

[11] D. Tian and N. D. Georganas. A coverage preserving node scheduling scheme for large wireless sensor networks. In *ACM WSNA*, 2002.

[12] Y. C. Wang, C. C. Hu, and Y. C. Tseng. Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In *IEEE WICON*, 2005.

[13] T. Yan, T. He, and J. Stankovic. Differentiated surveillance for sensor networks. In *ACM SenSys*, 2003.

[14] D. Yau, N. Yip, C. Ma, N. Rao, and M. Shankar. Quality of monitoring of stochastic events by periodic and proportional-share scheduling of sensor coverage. In *ACM CoNext*, 2008.

[15] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Journal of Wireless Ad-hoc and Sensor Networks*, 2005.